2)

```
import collections

def main():

    starting_node = [(0,0)]
    jugs = get_jugs()
    goal_amount = get_goal (jugs)
    check_dict = {}
    is_depth = True

    Search (starting_node, jugs, goal_amount,
          check_dict, is_depth)


def next_transitions (jugs, path, check_dict):

    Print (" finding next transitions & checking
           for the loops...)

    result = []
    next_nodes = ()
    node = []
```

1

ffb

```
a_mox = jugs (0)
b - mox = jugs (1)

a = Path [-1] [0]
b = Path [-1] [1]

node.append (a)
node.append (b_mox)
if not been_there (node, check_dict):
    next_nodes.append (node)
    node = []


node.append (a)
node.append (b_mox)
if not been_there (node, check_dict):
    next_nodes.append (node)
    node=()


node.append (min (a_mox, a+b))
node.append (b - (node[0] - a))
if not been_there (node, check dict):
    next_node.append (node)
    node = ()
```